Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

02255255.8

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 02255255.8
Demande no:

Anmeldetag:
Date of filing: 26.07.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

BRITISH TELECOMMUNICATIONS public limited company
81 Newgate Street
London EC1A 7AJ
GRANDE BRETAGNE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method and apparatus for locating devices

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G01S3/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR

Method and Apparatus for locating devices

Technical Field

The present invention relates to a method and apparatus for locating devices. It particularly relates to locating relatively simple and inexpensive devices which have the ability to communicate with one another in a wireless manner, provided the devices are located sufficiently close to one another to be "in range", when such devices are located with a sufficiently high distribution density that the devices are able to form an "ad-hoc" network by which the devices may communicate with one another and with external devices having an appropriate connection to the wireless ad-hoc network.

Background to the Invention and Prior Art

Wireless ad-hoc networks, which are self-organising, rapidly deployable and which require no fixed infra-structure since they are made solely (or at least largely) of self-contained wireless devices, are known and have been the subject of considerable research in recent times. However, such networks as proposed heretofore have been restricted in their approaches to obtaining positional information about individual devices within the network. In the main, three different types of approach have been adopted: approach 1 is simply not to provide positional information about individual devices within the network; approach 2 is to rely on such information being known *a priori* (eg by having been carefully placed in pre-ordained positions) and not being liable to change in an unpredictable manner; approach 3 is to use devices which include Global Positioning Systems (GPS's). In certain applications for such networks however, none of the above approaches is ideal. For example, in an application for sensing ocean conditions using an ad-hoc network of free floating sensors the first approach is not ideal because the sense data from each sensor is only valuable if accompanied with the position of the sensor at the time the data was recorded; the second approach is not tenable because the sensors will move with the ocean currents in an unpredictable manner (even assuming their initial placement was known); and the third approach is non-ideal because of the expense of supplying each sensor with a GPS.

In "GPS-free positioning in mobile ad hoc networks" by Srdjan Capkun, et al., Cluster Computing, Volume 5, # 2, April 2002 the authors describe an algorithm for permitting devices within a wireless ad-hoc network to obtain relative positional information without the use of any GPS containing devices, using only information about the distances between devices in range of one another (which information, it is said, can be found using a Time of Arrival (TOA) or similar range finding method). The paper mentions that the relative positional information could be associated with a geographical coordinate system only "if the algorithm is used along with some GPS-capable devices." However, it does not mention how this would be done. It also points out that for some applications (of particular concern to the authors) purely relative positional information is sufficient. The main drawback with the proposed method is that it does not scale well to large systems. The number of communications that each node is required to make increases as the number of nodes in the network increases. This means that beyond small networks the time needed and the processor power required becomes restrictive. Additionally, the calculations required of each device to execute the described algorithm are relatively arduous for very simple devices and therefore likely to be costly in terms of power consumption; additionally, the complexity has a negative impact on the speed with which the relative positions of the devices can be recalculated in the event of movement of the devices.

An alternative approach has also been considered in the field of ad-hoc wireless networks which, however, involves the use of base-stations for location purposes. In this approach (which may be thought of as a semi-ad-hoc semi-cellular approach) simple devices are able to communicate with one another to navigate data through the network (rather than just communicating with base stations as in purely cellular systems), but use only the base stations for location determination purposes (ie they do not attempt to determine their location from the locations of neighbouring simple devices only from neighbouring base stations). This approach has the disadvantage that a relatively large number of the more expensive base stations are required throughout the network.

Summary of the Invention

According to a first aspect of the present invention, there is provided a method of obtaining positional information about individual wireless devices within a wireless ad-hoc network including a plurality of position determining devices in which each position determining device includes means for estimating the distance between itself and any other similar device forming part of the network which is within range, the method including the steps of:

i) each position determining device receiving a broadcast message from each other similar device in range specifying, if known, the respective broadcasting device's position or an estimated position;

ii) each position determining device attempting to measure its distance from each other similar device in range;

iii) each position determining device determining its actual position or an initial estimated position and storing this information;

iv) each position determining device which does not know its actual position calculating the hypothetical distance between its estimated position and the position or estimated position of each neighbouring device whose broadcast position or possible position has been received and whose distance from the respective node has been measured in step ii);

v) each position determining device which does not know its actual position comparing the hypothetical distance calculated in step iv) with the distance measured in step ii);

vi) each position determining device which does not know its actual position, modifying its estimated position so as to reduce an error function dependent upon the difference between the hypothetical and measured distances and

vii) each position determining device broadcasting to each other similar device in range, if known, its actual position determined in step iii) or its modified estimated position determined in step vi).

This method works very well where the ad-hoc network includes a small percentage of devices which have some initial knowledge about their position. This may, for example, be achieved either by having some devices placed in fixed known locations or by having some devices which include a GPS. Since only a small fraction of the

total devices used need to be specially treated in this way, there is only a very small overall cost increase compared to a network in which none of the devices are specially treated in this way.

Note that steps (i) and (ii) may be performed in any order (or even simultaneously or in an overlapping or interleaved way, and may be different for different devices).

Preferably, the above described steps are repeated on a regular basis. However, in some applications the devices will remain in fixed locations and it is not therefore necessary to repeat all of these steps at each repetition or iteration. For example, in such a circumstance the relative distances between neighbouring nodes would only need to be determined once. Also, once a device has identified its actual position it will only need to repeat the final step (vii), and this only until all of its neighbouring devices have established their own locations, or the network decides that no further modification of estimated positions can result in a stable low error solution being found because of the particular distribution of the devices within the network.

Thus one embodiment of the invention takes a wireless ad-hoc network of nodes that each have a set range of communication. Within this network some nodes have information about their location (it is envisaged that each of these is either carefully placed and fixed at a certain location or belongs to a small sub-set of more expensive devices each of which is equipped with a GPS receiver). The nodes in the network then determine their distance to other neighbouring nodes that are within range of communication (exactly how would depend on the environment that they are in but something along the line of sonar or radar is envisaged). It is possible for a node that does not know its location, but is within range of at least one other node that does know its location (or an estimated location), to determine that it is located at some point on the circle surrounding the neighbouring node whose radius is equal to the measured distance from the respective neighbouring node. As an initial estimate (or guess) the node randomly picks any point on the circle. This initial estimate is then iteratively modified according to an algorithm which seeks to minimise the difference between the measured distances between neighbouring nodes and the hypothetical distances calculated using the current estimations of the positions of the nodes. The

present embodiment thus allows nodes to effectively locate themselves relative to each other in a very low-cost fashion, and then use partial information about their absolute position to enable a proportion of nodes in the network (seemingly dependent on the density of nodes in the network) to locate themselves accurately and absolutely. It applies to both fixed (non-moving) and fully mobile networks equally, with mobile networks effectively re-calculating the nodes' positions at regular intervals.

Preferably, the network includes at least three devices which have *a priori* knowledge about their position (or their possible positions). Note it is not necessary that at least two devices which have *a priori* knowledge about their position are located sufficiently close together for both devices to be simultaneously in range of a third device.

In one preferred strategy for distributing the devices of the network, there are a small number (approximately 5-10%) of 'gps' devices (ie devices having *a priori* positional information) that are either in planned locations or are distributed so that there are very few redundant devices giving excess information to nodes that don't need it. In another preferred strategy, there is a larger number (maybe as high as 25%) of gps devices that are completely randomly distributed in exactly the same way as the other non-gps devices. In both cases there is a degree of trade-off, the denser the network (i.e. the greater the number of devices that have no information about their initial location) the lower the number of 'gps' devices that are required.

In a preferred embodiment, all of the devices are position determining devices in the sense that they will each attempt to identify their position based on the information broadcast to them from neighbouring devices in the network together with range information determined using whatever distance detection mechanisms they possess. However, the possibility of including certain devices which form part of the network but which make no effort to determine their own position is not excluded. Similarly, for the purposes of this specification, devices which have some *a priori* knowledge about their position (and thus do not need to determine their position according to

the above set out method) are intended to be covered by the term "position determining devices."

In one embodiment, when each device first generates an estimated position, if it has information from more than one neighbouring device, it will use information from the neighbouring nodes according to the following order of priority: neighbouring nodes which say they know their actual position have highest priority, thereafter neighbouring nodes, which have broadcast an estimated position, have priority according to the measured distance of the neighbouring node from the respective determining node, with closer nodes having higher priority than more distant nodes. Where it is possible for an estimated position to satisfy the constraints set by more than one neighbouring node (ie where the circles, which are centred around the respective broadcast positions of the neighbouring nodes and have radiuses equal to the measured distance to the respective neighbouring nodes, coincide or overlap in one or more positions), then such an estimated position will be selected as the initial estimated position. In certain embodiments, when a device, after having already determined an initial estimated position (and possibly also having modified the initial estimated position one or more times), subsequently receives a broadcast of an actual or estimated position from a neighbouring node which is within range and which has a higher priority than the node or nodes whose broadcast information was originally used to determine its initial estimated position, it re-determines its "initial" estimated position using the information newly received from the higher priority node.

Preferably, the error function, which is dependent upon the difference between the hypothetical and measured distances, is dependent upon the sum of the squares of the differences between each measured and each hypothetical distance between a respective node and its nearest neighbours (ie for a node i having near neighbours 2, 3 and 4 the error function which node i seeks to minimise is:

$$E = \sum_{j=2}^{4} \left(md_{ij} - hd_{ij}\right)^2$$

Where md_{ij} is the measured distance from node i to node j and hd_{ij} is the hypothetical distance from node i to node j).

Preferably, each device includes a resetting algorithm the purpose of which is to avoid the device from getting its position estimation "stuck" in a local minimum of the error function. Such local minima in the error function often occur at locations approximately symmetrically opposite the actual position of the device with respect to a line connecting together two neighbouring nodes with respect to which the device is attempting to position itself. Preferably therefore, the resetting algorithm causes the reset estimated position to be approximately a mirror reflection of its current estimated position with respect to a line connecting the broadcast (actual or estimated) positions of the closest pair of neighbouring nodes which have broadcast their (actual or estimated) positions.

Preferably, the resetting algorithm considers one or more of the following parameters when deciding whether or not to reset its estimated position: the value of the error function calculated in respect of the current estimated position, both when compared to the average value for the error function during all preceding iterations and as an absolute value, the number of times the estimated position has been modified without resetting and how recently a neighbouring node has performed a reset.

By analogy with the well known technique of simulated annealing, the present procedure can be thought of as a type of local-knowledge-only simulated annealing technique in which the algorithm can be considered as minimising the energy of the system (where the energy of the system is the sum of the energies of the individual components and each individual component's energy depends, in the present example application, on how far away from its real position it thinks it is). The difference between this method of the present invention and conventional simulated annealing is that this method does not require a global measure of total energy to be available.

As an example of a very different field to which this procedure could be applied, consider the following problem:

There is a set (large) number of stars in certain positions to be photographed. A telescope must decide what is the best path between stars so that every star is photographed and the minimum total "distance" is travelled. This is a version of the well known 'travelling salesman' problem. Basically the approach using the prpesent procedure would be, pick a star, go to that star, photograph it, pick another star that hasn't been photographed, repeat. What the procedure would do would be to minimise the distance (energy) between successive stars. i.e. the algorithm would choose the star nearest to it which hadn't already been photographed. In this way the algorithm would have no knowledge of the global energy (i.e. the total distance travelled), but only the local energy (i.e. distance from star A being photographed to star X). The reset clause would come in where a star was reached and it had no stars within a suitable distance to move on to, at that point it would be returned to the previous star in the sequence and told to select a different star.

Further preferred features of the first aspect of the present invention are set out in the claims dependent upon claim 1.

According to a second aspect of the present invention, there is provided a method of operating an individual device as set out in claim 10.

According to a third aspect of the present invention, there is provided a device as set out in claim 11.

Brief Description of the Drawings

In order that the present invention may be better understood, embodiments thereof will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic diagram of an arrangement of position determining devices some of which have *a priori* knowledge of their position and the others of which perform an iterative algorithm to find estimations of their positions;

Figure 2 is a schematic diagram of the arrangement of Figure 1 illustrating the estimated positions of each of the nodes not having *a priori* knowledge of their position after approximately 1300 iterations of a position estimation procedure according to the present invention;

Figure 3 is a flow chart of the iterative procedure performed by the devices illustrated in Figures 1 and 2 not having *a priori* knowledge of their position;

Figure 4 is a flow chart illustrating the sub-steps of a "CONSIDER RESET" sub-routine of the procedure of Figure 3;

Figure 5 is a contour graph of the error function versus position of a single node communicating with three neighbouring nodes;

Figure 6 is a three dimensional representation of the contour graph of Figure 5;

Figure 7 is a plot of the noise function used to generate the graph of Figure 9;

Figure 8 is a graph showing the increase of the percentage of nodes which correctly estimate their position as a function of the total number of "random" nodes in the arrangement (having 16 nodes having *a priori* knowledge of their position); and

Figure 9 is a graph similar to that of Figure 8 showing the effects of noise introduced to the measurement of distance between neighbouring nodes.

Detailed Description of Embodiments

Referring firstly to Figure 1, one aspect of the present invention is an algorithm or procedure by which individual devices forming nodes within an ad hoc network can attempt to estimate their position based on the information broadcast to them by their neighbouring nodes together with a measured value of their distance from those neighbouring nodes. A first example of such a procedure, described below, is

particularly advantageous because it is robust to noise within the measurements of distance between nodes and because it is effective even in ad hoc networks in which a small number of nodes having *a priori* knowledge of their position (henceforth referred to as "*a priori*" nodes) are so thinly distributed that no two such nodes are sufficiently close to one another for both nodes to be able to communicate (simultaneously) with a third node (not having *a priori* knowledge of its position).

In order to demonstrate the robustness of the example procedure in an ad hoc network with a sparse density of "*a priori*" nodes and some noise on the distance measurements, the operation of the example procedure in an ad hoc network having the arrangement shown in Figure 1 is described below (note the actual performance of the procedure was carried out on a computer simulation). The particular arrangement of Figure 1 was formed by providing a grid of sixteen "*a priori*" nodes a1,a2,...,a16 evenly spaced apart from one another such that their ranges of communication a1r, a2r,...,a16r just touch one another without any overlap, as shown. On to this grid area (a square having sides equal to four times the diameter of each range of communication circle a1r – a16r), 125 nodes r1,r2,...,r125 which do not have any "*a priori*" knowledge of their position are randomly distributed (for this reason they are henceforth referred to as random nodes as distinct from the "*a priori*" nodes which are placed in non-random specified positions).

The position estimation procedure is then carried out until a reasonably stable solution is arrived at. This may be determined on a node by node basis by simply freezing the node's individual position estimation if it has not changed significantly for a predetermined large number of iterations. However in the present simulation, the procedure was simply allowed to continue until no further significant improvement was being achieved in the overall percentage of nodes successfully locating themselves within 5% of their correct position. This occurred after approximately 1300 iterations of the procedure. The situation after these 1300 iterations is illustrated in Figure 2 in which the actual position of each random node which has managed to form an estimation of its position (124 out of the 125 random nodes) is marked with a hollow square and the estimated position of each node is indicated by a hollow circle (which, if it is located outside the corresponding hollow

square, is connected to its respective hollow square by a straight line). The single random node r70 which was not able to locate itself is located towards the centre of the grid and was unable to locate itself at all because it was not in range of any other node (note that the random nodes have exactly the same range of communication as the *"a priori"* nodes though these are not marked in either Figure 1 or 2 to avoid rendering the Figures unclear due to excessive clutter).

Note that in the description of the procedure below, details of how and when the positional information from neighbouring nodes is received, the details of how and when the current estimate is broadcast to other nodes and details of how and when the distance to neighbouring devices is measured are deliberately not discussed. This is because these procedures are considered to operate at a different modular "layer". Operating in this way is advantageous because the details of how these lower level procedures are performed should not be of concern to the procedure described below. Instead, an interface is provided whereby the procedure is able to access the latest positional information which has been broadcast to the device, the latest measured distances and it makes available at all times the current estimated position, from and to the lower layer respectively, which is then responsible for communicating with its neighbouring devices in a conventional manner. All that is required is that the communications should happen at the same frequency with which iterations are performed. Preferably, the communications with neighbouring nodes and the measurement of distances to neighbouring nodes occurs when the procedure described below is waiting between execution of consecutive iterations. However, if the network is static, it is not necessary to continually update the distance measurements since these will not change.

Referring now to Figure 3, the procedure followed by each random node is described below. Note that the *"a priori"* nodes need only broadcast their (assumed to be completely accurate) positions at the beginning of each iteration. Prior to commencing the procedure, each device will have a few variables etc. initialised including initialising an estimate-made flag, indicating whether or not a first estimate at the device's position has yet been made, to an unset state, initialising a first-average-calculate flag, which indicates whether the first "average" error has still to

be calculated, to a set state and initialising the number of iterations of the procedure to zero and a No-of-guesses variable (whose purpose becomes clear from the following description) to zero. With these initialisations made, the procedure can commence with flow passing from the start of the flow chart to step S5 in which the device gets and stores positional information (if any is available) received from neighbouring nodes which are within communication range of the specified device. Note that to render the description of the procedure as clear as possible, the procedure will henceforth be described from the point of view of a single device hereinafter referred to, where necessary for the sake of clarity, as the present device.

The positional information is broadcast between the nodes using a limited range radio communication, the precise details of which are not relevant to the present invention and will not therefore be discussed here in detail. As will be apparent to a person skilled in the art, the communications between the devices can be carried out in a number of well known manners. 'Ad Hoc networking', Charles E. Perkins, Pub. Adison Wesley, Dec. 2001 and the references cited therein provide a detailed description of some suitable such methods.

Upon completion of step S5, flow passes to step S10 in which the present device gets and stores distance information (ie the measured distance between the present device and each neighbouring node within communication range).

As with methods of broadcasting information between nodes, there are many well known methods of determining the distance from one device to another which are suitable for use within an ad hoc network. For example, the Radio Acoustic Ranging (RAR) method or a similar method based on detecting the difference in time taken for a sound wave versus a radio wave to travel the distance to be measured (or a known multiple thereof). would be appropriate. Implementations of these method are well known and will not be discussed here in detail since they are not germane to the present invention. Chapter 6 of 'Hydrographic Manual' by Commander K. T. Adams, Special publication of US dept. of commerce, Coast and geodetic survey published by US Govt. printing office, 1942 and the references cited therein provide a detailed description of RAR.

Upon completion of step S10, flow passes to step S15 in which it is determined whether or not a previous estimate of the present device's position has been made during an earlier iteration of the procedure. In the present embodiment, this is determined by seeing if the estimate-made flag is unset or set. If the flag is still unset, flow passes to step S20 where it is determined if the device has sufficient information to make a first estimate of its position.

To do this, it needs to know the actual or estimated position of, and measured distance to, at least one neighbouring node (henceforth the term "neighbouring node" will be used to refer to a node which is within communication range). If this information is known an estimate of the device's position will be made based on the information. If it has this information in respect of more than one neighbouring node, it will base its estimate on the information of the highest priority node (with *"a priori"* nodes having highest priority or otherwise the closest random node).

As an example, on the first iteration, node r1 (together with all of the other nodes within range of the *"a priori"* node a1) will receive the necessary information only from *"a priori"* node a1, since none of its other neighbours (eg node r2) will yet have had a chance to estimate their position. Thus node r1 will estimate its position as being anywhere on the circle whose centre is given by the broadcast position of the *"a priori"* node a1, and whose radius is equal to the measured distance (as measured by device r1) between device r1 and *"a priori"* node a1. To specify a particular position, the device chooses a random number between 0 and 2π and uses this to set an angle with the direction along the x-axis in the positive direction corresponding to zero (radians), for example. By contrast, random node r9 is not within range of any *"a priori"* nodes and thus on the first iteration will not have any information on which to estimate its location. However, by the second iteration its random node neighbours (r5, r7 and r8) will all have formed an estimate of their position (based on node a1) and will have broadcast this to node r9. Since random node r8 is the closest of these three, the estimated position from this node, and the measured distance therefrom, is used to form an estimate of the position of node r9 (again by choosing a random angle between 0 and 2π radians and using this to select a

position on the circle centred on the estimated broadcast position of node r8 and having radius equal to the measured distance - as determined by node r9 - between nodes r9 and r8).

If at step S20 it is determined that there is insufficient information to form an estimate of its location, the iteration is ended and flow is returned back to step S5 where the variable storing the number of iterations of the procedure is incremented and a predetermined time is allowed to elapse before performing a further iteration. Note that an iteration comes to an end and the flow is passed back to step S5 to await the commencement of a new iteration, the variable storing the number of iterations of the procedure is incremented and a predetermined time is allowed to elapse before performing the next iteration.

If at step S20 it is determined that there is sufficient information available to make a first estimate, then flow is passed to step S25 where a first estimate is made in the manner described above (ie using the highest priority neighbouring node's broadcast position and the measured distance to the neighbouring node and randomly choosing a point on the appropriate circle), and the estimate-made flag is set. Then the current iteration is ended and flow is returned to step S5.

If at step S15 it is determined that the estimate-made flag is set, then the flow is passed to step S30, in which the perceived error, E, for the estimated position determined by the present device in the previous iteration is calculated. This error is calculated on the basis of the most recently broadcast positions of the present device's neighbouring nodes and the measured distances from the present device to each neighbouring node using, in the present example, the following equation:

$$E = \sum_{All\_Neighbouring\_Nodes} (md - hd)^2$$

where md is the measured distance between the present device and a given neighbouring node, and hd is the hypothetical distance between the present device and the given node.

Upon completion of step S30, flow passes to step S35 in which it is determined whether or not an "Average" Error has yet been calculated. In the present embodiment, this is done by examining the first-average-calculate flag to see if it is set or unset. If the flag is set (indicating that the first calculation of an "Average" Error is yet to be performed), then flow is passed to step S40.

In step S40, the "Average" Error is set to equal the value calculated for the perceived Error , E, calculated in step S30, the first-average-calculate flag is unset and flow is passed to step S50.

If in step S35 it is determined that an "Average" Error has previously been calculated then flow is passed to Step S45 where the "Average" Error is up-dated. This is done using the formula "Average" Error = ("Average" Error + perceived error E)/2.

Upon completion of step S45, flow is passed to step S50 in which it is determined if the newly calculated perceived error, E, is greater than the newly calculated "Average" Error. If it is not, it is deemed to be indicative of a general trend towards an improved estimated position and flow is passed to step S55.

In step S55, a new estimated position is calculated which has a reduced perceived error . In the present embodiment, this is done in the following manner:

Firstly, the gradient of the perceived error is calculated (ie $\underline{\nabla}E$ is calculated) at the current estimated position of the present device to determine the direction in which the perceived error falls most quickly. A first trial position is then selected along the line extending from the current estimated position of the present device, the distance from the current estimated position of the trial position being given according to a predetermined formula which in the present embodiment is given by:

$$TRIAL\_POS = Const.|\underline{\nabla}E|$$

Where Const is a predetermined constant which is set in dependence upon the particular geometry of the ad hoc network arrangement and in the present example is set to a value of sixteen distance units. Each node now has stored 2 values of E at either end of the line of Del(E). An iterative procedure is then followed which comprises the following steps: pick another point on the line of Del(E) halfway between the previous points and calculate the perceived error, E, at that point; discard the highest value error point (one end) and repeat. In the present example, this iterative procedure is followed for a total of 10 times. The procedure is, in essence, a Newtonian approximation which is known to quickly home in on a minimum value in certain circumstances (where there are not lots of local minima). It is used in the present example because the error function, E, generally doesn't have a huge number of local minima along a line of Del(E). It is possible that the approximation will find a local minima along the line rather than the global minimum, but it is unlikely, and if it does the procedure still functions adequately. Also as part of this step, the No-of-guesses variable is incremented by one.

Upon completing step S55, the current iteration is ended and flow is passed back to step S5.

If at step S50 it is determined that the perceived error is greater than the "Average" Error, this is taken as a sign that the present device may be stuck in a local minimum of the perceived error, E, which is actually some distance removed from the actual position of the present device. As a result, flow is passed to subroutine S60 at which it is determined whether or not the present device should perform a reset, hold its current estimated position or generate a new estimated position in the normal way (ie via step S55). The steps performed within subroutine S60 are described in greater detail below with reference to Figure 4.

Upon completion of subroutine S60, flow proceeds to step S65 in which it is determined whether subroutine S60 decided that a "reset" should occur. If it was

not decided to reset, flow proceeds to step S70 in which it is determined whether subroutine S60 decided that a "hold" should occur. If it was determined that a "hold" should occur, then the iteration is ended and flow returns to step S5. If it is determined in step S60 that a hold should not occur, then flow proceeds to step S55 in which a new estimate is formed in the manner described above.

If at step S65 it is determined that a "reset" is to occur, flow proceeds to step S75. In step S75 a new estimate of the position of the present device is made according to the following procedure, in the present example: one of the neighbouring nodes is selected and the new estimate is set to be at a position the same distance from the selected node but on the opposite side (ie the new estimated position is a rotation about the selected neighbouring node of π radians from the former estimated position). Additionally in this step, the No-of-guesses variable is reset to zero.

Upon completion of step S75, the current iteration is ended and flow returns to step S5.

Referring now to Figure 4, at the commencement of the subroutine S60, flow passes to step S105 in which the current value of the perceived error, E, is compared with a predetermined threshold, E_{THRESH}, which, in the present example, is set to equal four times the number of neighbouring nodes to the present device. Note that optimally the threshold for perceived error is dependent on the level of expected noise (more noise, higher threshold). It has been found that for an approximate expected noise of 5%, a threshold of 1x(# of neighbouring nodes) works well whereas for 10% noise (which corresponds approximately to the amount of simulated noise used in the present example), a threshold of 4x(# neighbouring nodes) was found to work well. Note that 10% noise can be taken to mean that where the noise is approximately Gaussian, the vast majority of measured distances are within 10% of the actual distance which is trying to be measured. Note that it has been determined that the multiplication factor (1 in the case of 5% noise, 4 in the case of 10% noise) should optimally vary as the square of the expected percentage error in the system (i.e. doubling error raises the threshold by 4x).

If it is determined in step S105 that E is below the threshold E_{THRESH}, then flow proceeds to step S110 in which the resulting decision of the subroutine is set to indicate not to reset and not to hold and flow proceeds to the end of the subroutine.

The significance of this test is that if the perceived error is very low it is likely that the present device's current estimated position is in approximately the correct area and a reset would be likely to send the estimated position into the wrong area and therefore should be avoided.

If at step S105 it is determined that E is greater than E_{THRESH}, then flow passes to step S115.

At step S115 it is determined whether the value stored in the No-of-guesses variable, NO_OF_GUESSES, is below a threshold, $NO\_OF\_GUESSES_{THRESH}$, which, in the present example procedure is set to 20. If it is determined that NO_OF_GUESSES is below the threshold, $NO\_OF\_GUESSES_{THRESH}$, then flow proceeds to step S120 in which the resulting decision of the subroutine is set to indicate not to reset and not to hold and flow proceeds to the end of the subroutine.

The significance of this test is that if the number of guesses since the last reset (when the No-of-guesses variable is reset) is very low it is likely that the system is unstable and therefore it would be beneficial to allow the system to settle down a bit before attempting another reset. Additionally, immediately after doing a reset, it is likely that the perceived error will be much higher than before the reset. A few iterations should be permitted during which the estimated error should rapidly reduce as the estimated position moves towards a (hopefully) new minimum value.

If at step S115 it is determined that NO_OF_GUESSES is greater than $NO\_OF\_GUESSES_{THRESH}$, then flow passes to step S125.

At step S125 it is determined whether the value stored in a variable which stores the number of iterations elapsed since one of the present device's neighbours last reset, ITERATIONS_SINCE_NEIGH_RESET, is below a threshold value,

$ITERATIONS\_SINCE\_NEIGH\_RESET_{THRESH}$, which, in the present example procedure is set to 1 (note it has been found that this threshold value should be kept low, but above zero, for the procedure to run well, preferably between 1 and 5). If it is determined that the number of iterations since a neighbour last reset is below the threshold value, then flow proceeds to step S130 in which the resulting decision of the subroutine is set to indicate not to reset but instead to hold and flow proceeds to the end of the subroutine.

The significance of this test is that if a neighbour has recently reset, it is likely that the recently reset neighbour does not have an accurate estimate of its position. As a result, it is useful to give the recently reset neighbouring node an opportunity to move towards a minimum in the perceived error function (and hence hopefully to a position which is closer to its actual position) before attempting to adjust the position of the present device on the basis of the error function which is likely to be heavily influenced by the recently reset node causing a large increase in the perceived error determined by the present device.

If at step S125 it is determined that the number of iterations since a neighbouring node last reset is greater than the threshold amount, then flow passes to step S135.

In step S135, a random number is chosen (using a simple random number generator such as is commonly available on microprocessors) to lie between zero and a predetermined maximum value, MAX, the value of which depends upon the average distance between nodes in the network. In the present example, the maximum value is calculated according to the following formula:

$$MAX = RANGE^2 \div 2500$$

In the simulated example, the average distance between nodes in the network is approximately 50 distance units and so RANGE was set to 50 giving a value for MAX of 1. Having selected a random number between zero and MAX, flow passes to step S140 in which it is determined, in the present example, whether the chosen random number is less than the "Average" error divided by the number of guesses since the

present device was last reset (ie NO_OF_GUESSES) (note an alternative example could use some more complicated function of this ratio instead).

If the determination in step S140 is that the random number is not less than the "Average" error over NO_OF_GUESSES ratio, flow passes to step S145 in which the resulting decision of the subroutine is set to indicate not to reset but instead to hold and flow proceeds to the end of the subroutine. Otherwise, flow passes to step S150 in which the resulting decision of the subroutine is set to indicate to reset and flow proceeds to the end of the subroutine.

Note that the random number chosen in step S135 varies between 0 and 1 while the ratio of "Average" error over NO_OF_GUESSES varies potentially between 0 and a high number (ie much higher than 1). The significance of this is that when error is high, the present device will always reset, and this is desirable since high error is considered to be an indication of the present device failing to find a reasonable solution. It should also be noted that no reset will be made if the NO_OF_GUESSES is less than 20 since the subroutine would have terminated with a decision not to reset or hold after step S115.

Stated somewhat loosely, the overall effect of subroutine S60 is that if things are wrong then the present device will always reset, if given the chance (ie if the conditions specified in steps S105, S115 and S125 are satisfied such that flow proceeds to steps S135 and S140), if things are right the present device will never reset and if things are nearly right then the present device has a chance of resetting.

Figures 5 and 6 illustrate the way in which the perceived error function, E, varies with position, by way of a contour map in Figure 5 and a perspective view of a three-dimensional representation of the contour map in Figure 6. The arrangement of nodes on which the contour map is based is one where the node who's contour map is drawn is actually located at x=200, y=200 and it can communicate with 3 other nodes at (0,0), (200,0) and (0,200). The node at (0,0) is an *"a priori"* node that knows its location at all times, the other 2 are random nodes which have guessed their locations (incorrectly) to be (150,30) and (75,162) respectively. From this the

node who's contour map is shown will optimally guess a location which is in the region of lowest error (in the region around x=250, y=175). Depending on the exact type of procedure followed for arriving at the minimum, it can be seen that after a few iterations the guessed position would coincide with this global minimum. Note that because the random neighbouring nodes have incorrectly guessed their positions, the global minimum does not quite coincide with the node's exact correct position. However, it is expected that the neighbouring random nodes will improve their guesses of their location in subsequent iterations and in doing so, the global minimum will move towards the correct position.

Referring now to Figure 7, it was mentioned above that in the simulation described above in relation to Figures 1 to 4, a certain amount of noise (approximately 10%) in the measured distances was simulated. In the above described simulation, the noise was generated using the following function:

$$md = md_{true} \times R$$
$$R = (rand + rand + rand + rand)/20 + 0.9$$

where rand is a random number chosen from a uniform distribution between zero and one. Thus R is a random number between 0.9 and 1.1 having a distribution which is approximately Gaussian. Figure 7 illustrates a plot of 100,000 values of R chosen using the above formula. In an alternative simulation in which a random number generator function having a Gaussian distribution was available, this could be used instead, with perhaps a variance of 10% of the mean instead of maximum and minimum values of plus and minus 10% of the mean respectively.

Figure 8 shows the results of performing a large number of simulations with differing numbers of random nodes and in many different arrangements (all with the *"a priori"* nodes arranged in the same grid structure as shown in Figures 1 and 2 but with different numbers and random arrangements of the random nodes) all with zero noise on the measured distances. It can be seen that with only 75 random nodes, the performance is very variable with, on average, just under half of the random nodes being able to successfully locate themselves, with a best result of just under 80% of

nodes locating themselves successfully and a worst result of just over 10% locating themselves correctly. However, with 100 random nodes, the worst case is just under half of the nodes successfully locating themselves and the best case is exactly 90% with an average of just over 70%. At 125 random nodes (as in the example described in detail above with reference to Figures 1 to 4), the worst case is just over 70%, average is approximately 85% and best case is over 90%. Thereafter improvement is slower with increasing numbers of nodes until at 200 random nodes the worst case is about 95%, the average is just under 100% and the best result is 100% of random nodes correctly locating themselves.

Figure 9 illustrates the effects of adding noise to the measured distances. It can be seen that adding noise has a detrimental effect on the ability of the nodes to successfully locate them selves. However, there is only a small detrimental effect on the ability of the nodes to locate themselves to within 10 distance units of their correct locations. Given that the whole grid has a size of 400 by 400 distance units (ie the *"a priori"* nodes are spaced 100 distance units apart) this illustrates that the reduced performance caused by noise on the measured distances is not great. In fact, from the Figure, it can be seen that even with 10% noise on the average percentage of nodes able to locate themselves within 10 distance units of their actual position is only about 5% less than the average percentage of nodes able to correctly locate themselves when there is no noise.

Variations

An alternative method might utilise information from more distant nodes than just the nearest neighbour nodes. This information could be obtained by having all nodes periodically broadcast information about the position of their nearest neighbours in addition to information about their own position or estimated position. This information could perhaps be used to encourage a reset where a node cannot communicate with a node which has located itself in a position which should be in communication range if the node was where it currently estimated itself to be (ie the node can attempt to reset to a position further away from the next-nearest neighbour).

CLAIMS

1. A method of estimating the location of a device within a network of devices each of which forms a node of the network, the method including the steps of:

obtaining information specifying the location or estimated location of one or more neighbouring nodes;

measuring the distance to said one or more neighbouring nodes; and

iteratively modifying an estimated location of the device, such as to improve the consistency between the estimated location of the device and the location or estimated location of the one or more neighbouring nodes, as determined from the obtained information specifying the location or estimated location of the one or more neighbouring nodes, on the one hand and the measured distances to each of the one or more neighbouring nodes on the other hand.

2. A method as claimed in claim 1 wherein the consistency is determined according to an error function which depends on the sum of the square of the difference between a hypothetical distance to each neighbouring node and the measured distance to the respective node, where the hypothetical distance is the distance between the location or estimated location of the neighbouring node as obtained and the current estimated position of the device.

3. A method as claimed in either preceding claim wherein the network is a wireless ad-hoc network, and the method includes the step of wirelessly communicating with the one or more neighbouring nodes.

4. A method as claimed in any preceding claim further including the step of determining whether or not to reset the estimated location of the device if certain conditions are met to a location determined according to a procedure which does not seek to improve consistency, whereby the device can avoid getting its estimated location stuck in a local minimum value of consistency.

5. A method as claimed in claim 5 wherein the reset step includes a sub-step of probabilistically determining whether or not to reset wherein the probability of

resetting is reduced with increasing the ratio of the number of iterations since a reset was last performed to a measure of inconsistency, at least beyond a certain minimum and/or maximum value of this ratio.

6. A carrier medium carrying a computer program comprising processor implementable instructions for causing a device to carry out the method of any one of the preceding claims during implementation of the instructions.

7. A device for forming a node within a network of similar devices, the device including locating means for estimating its location, the locating means including:

obtaining means for obtaining information specifying the location or estimated location of one or more neighbouring nodes;

distance measurement means for measuring the distance to said one or more neighbouring nodes; and

processing means for iteratively modifying an estimated location of the device, such as to improve the consistency between the estimated location of the device and the location or estimated location of the one or more neighbouring nodes, as determined from the obtained information specifying the location or estimated location of the one or more neighbouring nodes, on the one hand and the measured distances to each of the one or more neighbouring nodes on the other hand.

8. A method of obtaining positional information about individual wireless devices within a wireless ad-hoc network including a plurality of position determining devices in which each position determining device includes means for estimating the distance between itself and any other similar device forming part of the network which is within range, the method including the steps of:

i) each position determining device receiving a broadcast message from each other similar device in range specifying, if known, the respective broadcasting device's position or estimated position;

ii) each position determining device attempting to measure its distance from each other similar device in range;

iii) each position determining device determining its actual position or an initial estimated position and storing this information;

iv) each position determining device which does not know its actual position calculating the hypothetical distance between its estimated position and the position or estimated position of each neighbouring device whose broadcast position or possible position has been received and whose distance from the respective node has been measured in step ii);

v) each position determining device which does not know its actual position comparing the hypothetical distance calculated in step iv) with the distance measured in step ii);

vi) each position determining device which does not know its actual position, modifying its estimated position so as to reduce an error function dependent upon the difference between the hypothetical and measured distances and

vii) each position determining device broadcasting to each other similar device in range, if known, its actual position determined in step iii) or its modified estimated position determined in step vi).

ABSTRACT

Method and Apparatus for locating devices

A method of obtaining positional information about individual wireless devices a1-a16, r1-r125 within a wireless ad-hoc network including a plurality of position determining devices a1-a16, r1-r125 in which each position determining device includes means for estimating the distance between itself and any other similar device forming part of the network which is within range. The method includes the steps of:

i) each position determining device receiving a broadcast message from each other similar device in range specifying, if known, the respective broadcasting device's position or estimated position;

ii) each position determining device attempting to measure its distance from each other similar device in range;

iii) each position determining device determining its actual position or an initial estimated position and storing this information;

iv) each position determining device which does not know its actual position calculating the hypothetical distance between its estimated position and the position or estimated position of each neighbouring device whose broadcast position or possible position has been received and whose distance from the respective node has been measured in step ii);

v) each position determining device which does not know its actual position comparing the hypothetical distance calculated in step iv) with the distance measured in step ii);

vi) each position determining device which does not know its actual position, modifying its estimated position so as to reduce an error function dependent upon the difference between the hypothetical and measured distances and

vii) each position determining device broadcasting to each other similar device in range, if known, its actual position determined in step iii) or its modified estimated position determined in step vi).

Figure 1

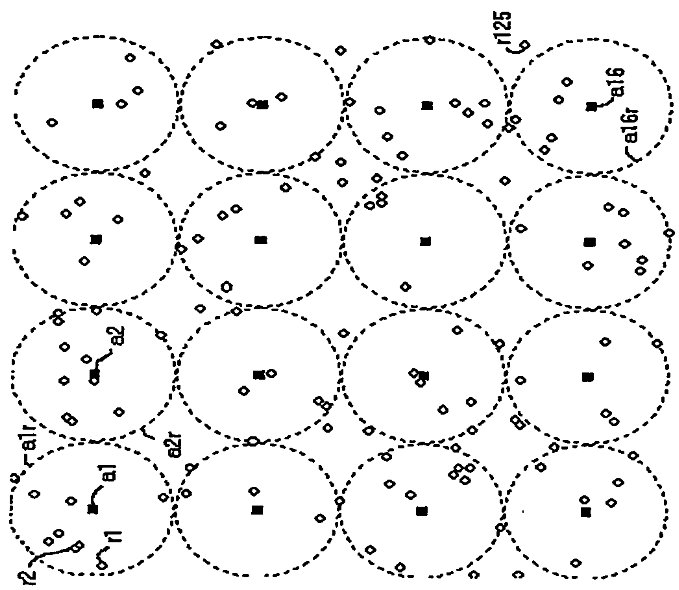


Figure 1

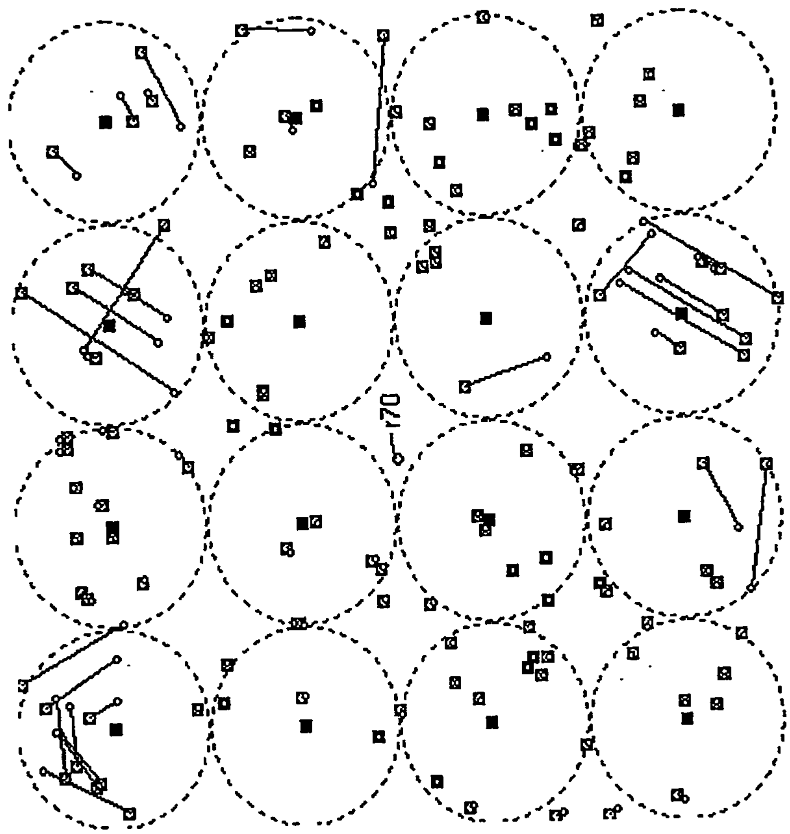


Figure 2

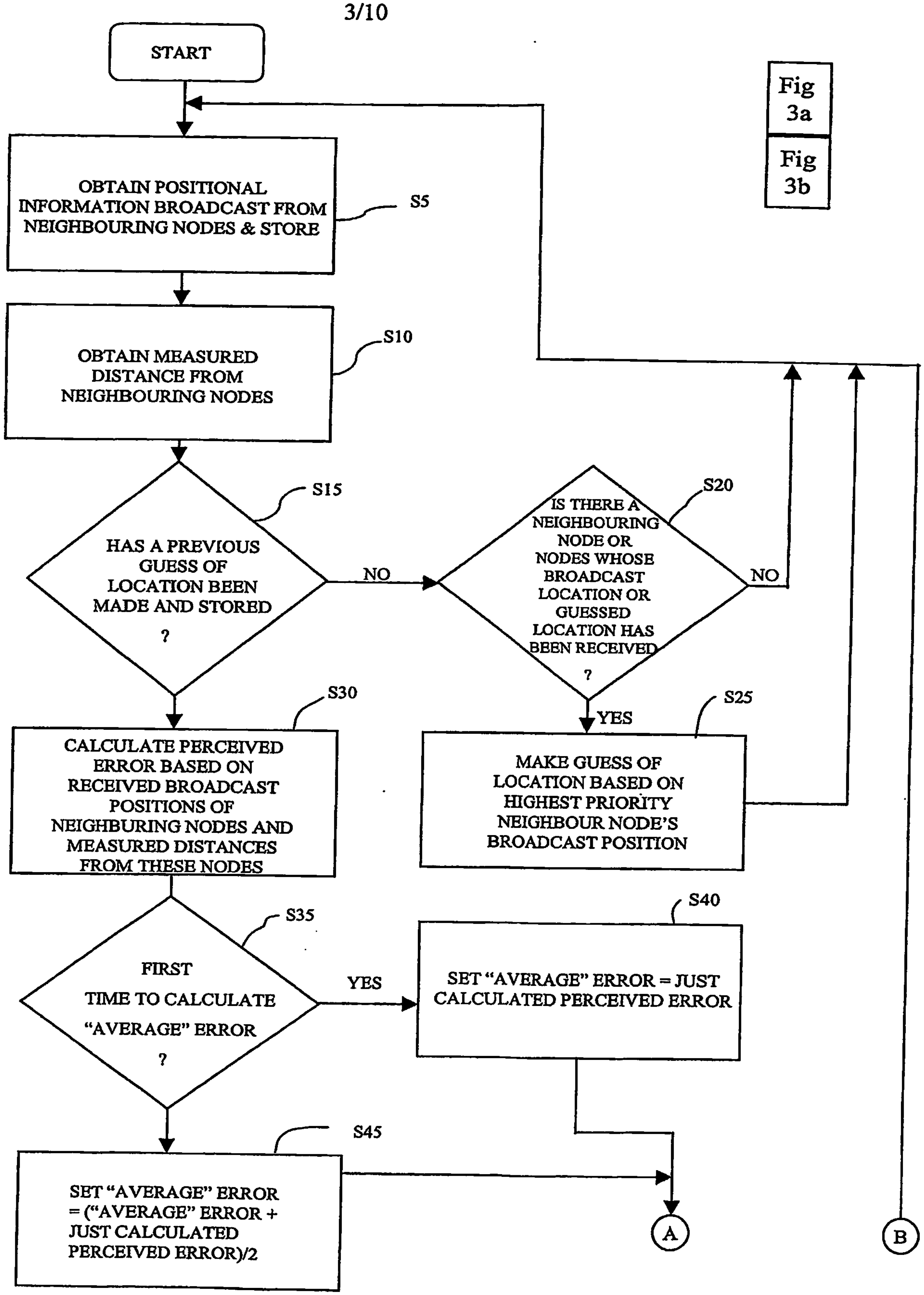
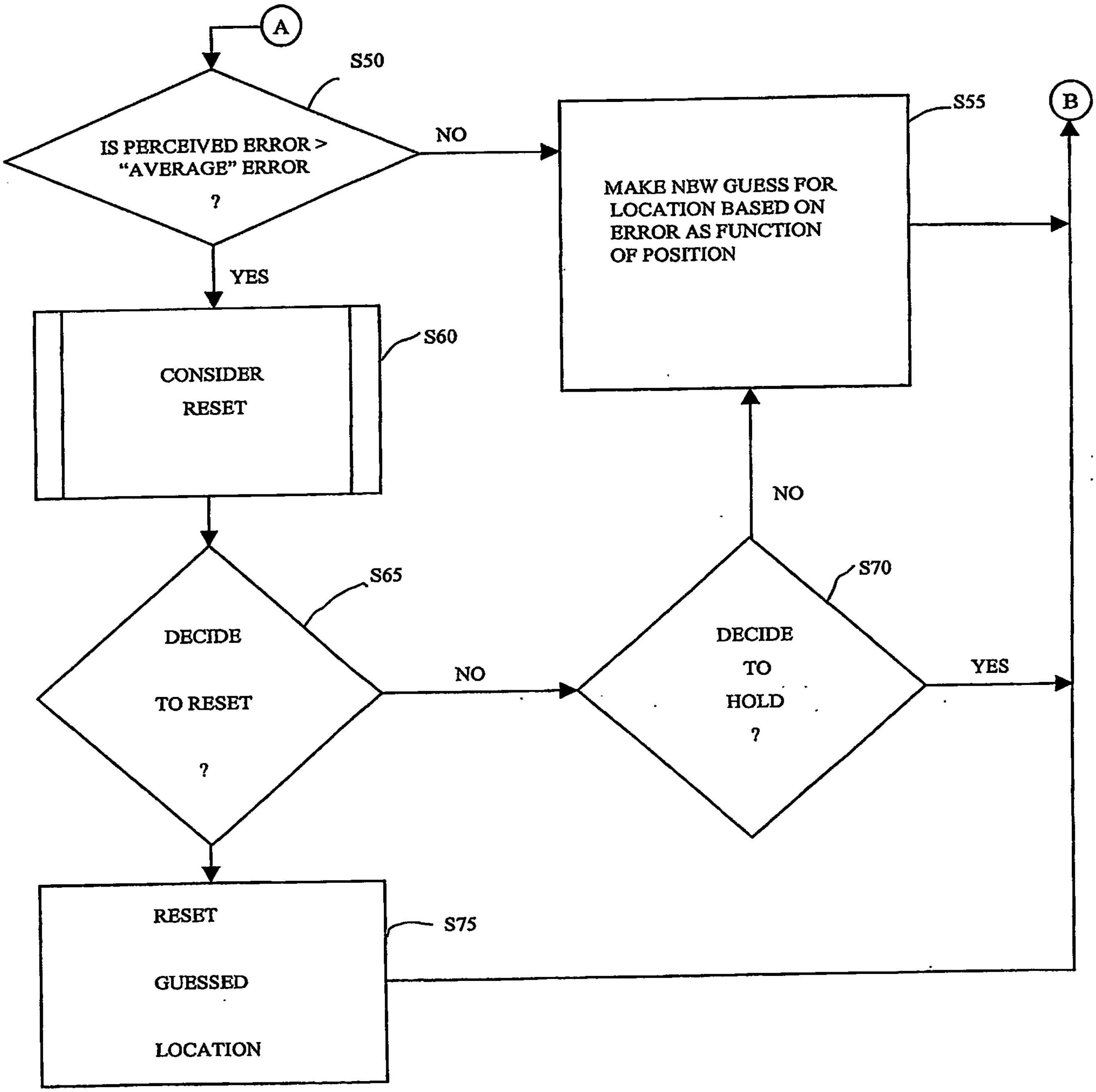

START
OBTAIN POSITIONAL INFORMATION BROADCAST FROM NEIGHBOURING NODES & STORE
S5
OBTAIN MEASURED DISTANCE FROM NEIGHBOURING NODES
S10
HAS A PREVIOUS GUESS OF LOCATION BEEN MADE AND STORED ?
S15
NO
IS THERE A NEIGHBOURING NODE OR NODES WHOSE BROADCAST LOCATION OR GUESSED LOCATION HAS BEEN RECEIVED ?
S20
NO
YES
S25
MAKE GUESS OF LOCATION BASED ON HIGHEST PRIORITY NEIGHBOUR NODE'S BROADCAST POSITION
S30
CALCULATE PERCEIVED ERROR BASED ON RECEIVED BROADCAST POSITIONS OF NEIGHBURING NODES AND MEASURED DISTANCES FROM THESE NODES
S35
FIRST TIME TO CALCULATE "AVERAGE" ERROR ?
YES
S40
SET "AVERAGE" ERROR = JUST CALCULATED PERCEIVED ERROR
S45
SET "AVERAGE" ERROR = ("AVERAGE" ERROR + JUST CALCULATED PERCEIVED ERROR)/2
A
B
Fig 3a
Fig 3b

FIGURE 3a



A
S50
IS PERCEIVED ERROR > "AVERAGE" ERROR ?
NO
YES
S55
B
MAKE NEW GUESS FOR LOCATION BASED ON ERROR AS FUNCTION OF POSITION
S60
CONSIDER RESET
NO
S65
DECIDE TO RESET ?
NO
S70
DECIDE TO HOLD ?
YES
RESET GUESSED LOCATION
S75

FIGURE 3b

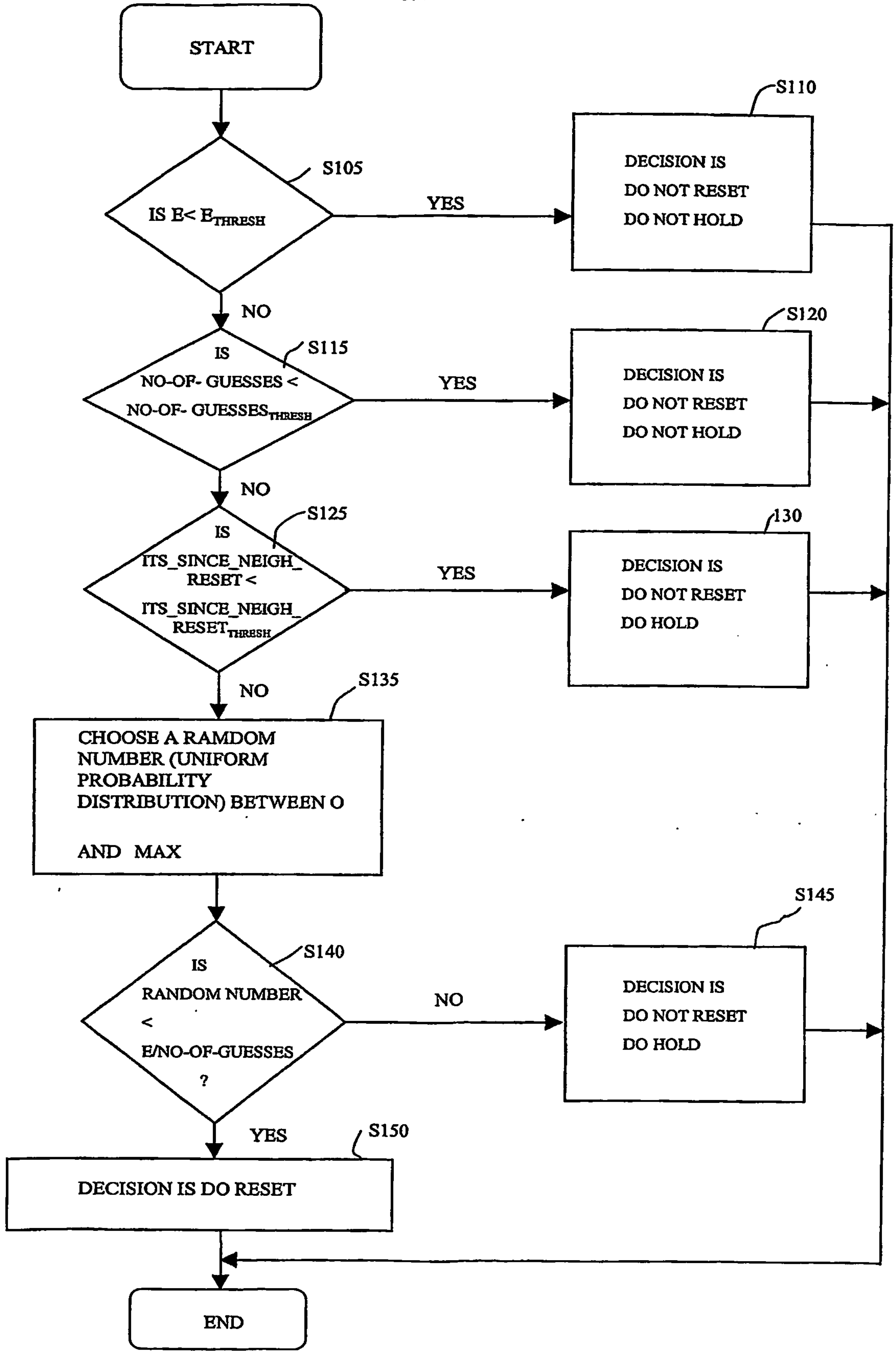

START
IS E< E THRESH
S105
YES
S110
DECISION IS
DO NOT RESET
DO NOT HOLD
NO
IS
NO-OF- GUESSES <
NO-OF- GUESSES THRESH
S115
YES
S120
DECISION IS
DO NOT RESET
DO NOT HOLD
NO
IS
ITS_SINCE_NEIGH_
RESET <
ITS_SINCE_NEIGH_
RESET THRESH
S125
YES
130
DECISION IS
DO NOT RESET
DO HOLD
NO
S135
CHOOSE A RAMDOM
NUMBER (UNIFORM
PROBABILITY
DISTRIBUTION) BETWEEN O
AND MAX
IS
RANDOM NUMBER
<
E/NO-OF-GUESSES
?
S140
NO
S145
DECISION IS
DO NOT RESET
DO HOLD
YES
S150
DECISION IS DO RESET
END

FIGURE 4

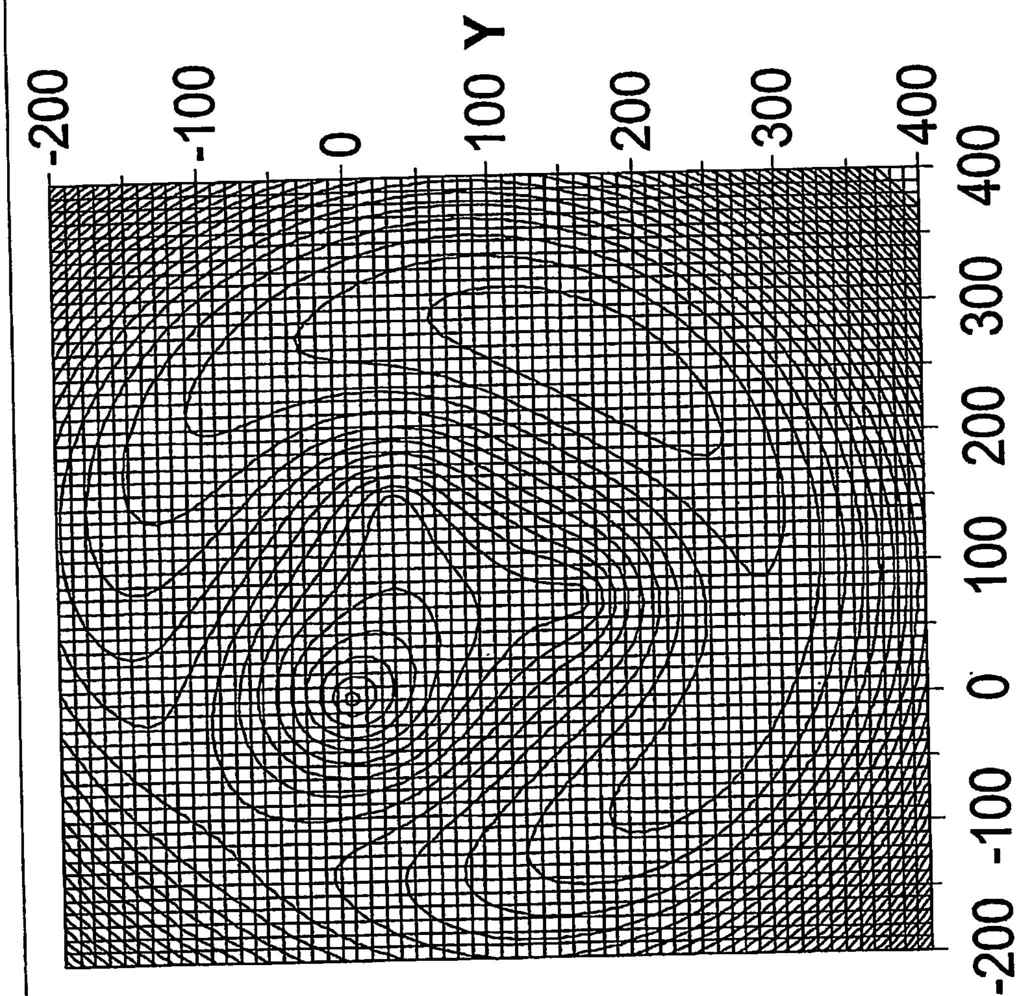


Figure 5

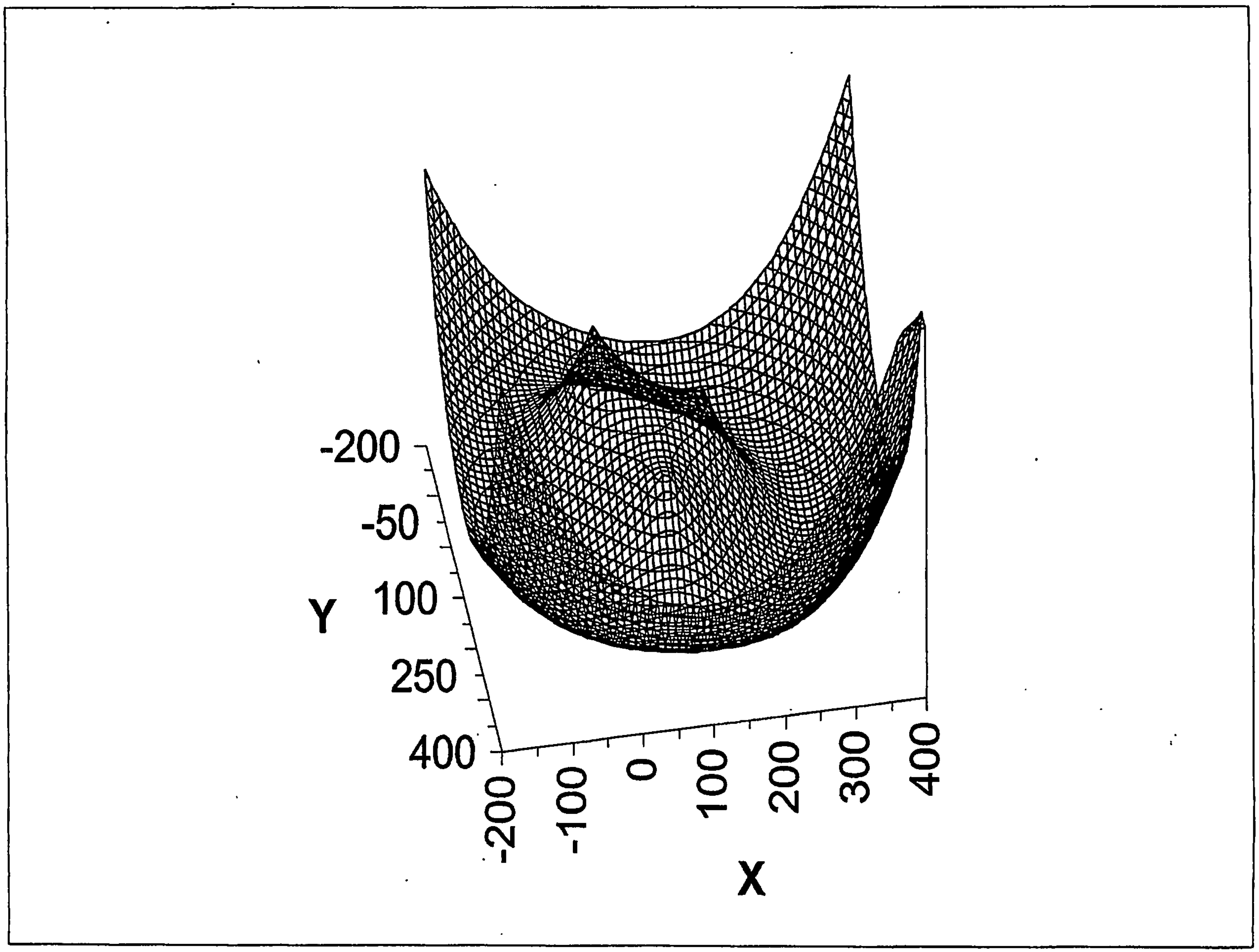


Figure 6

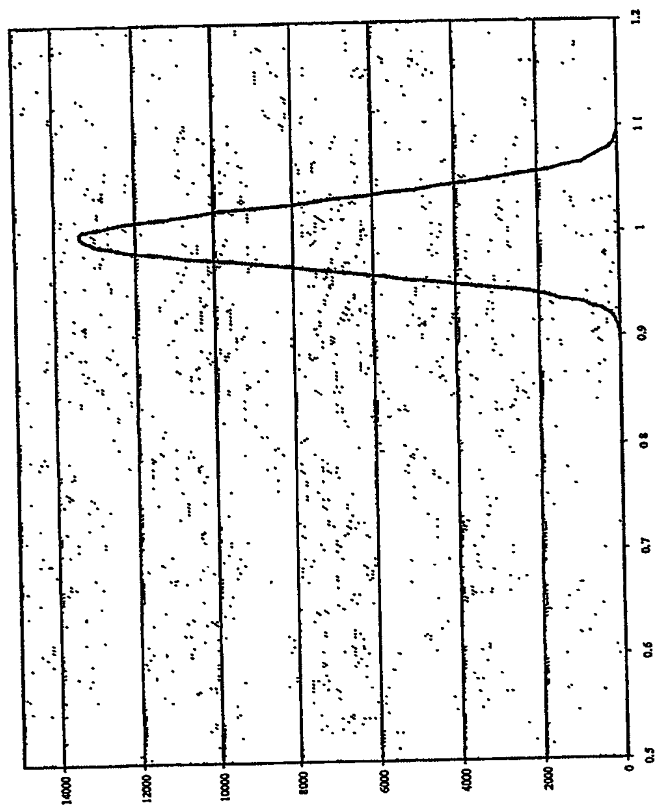
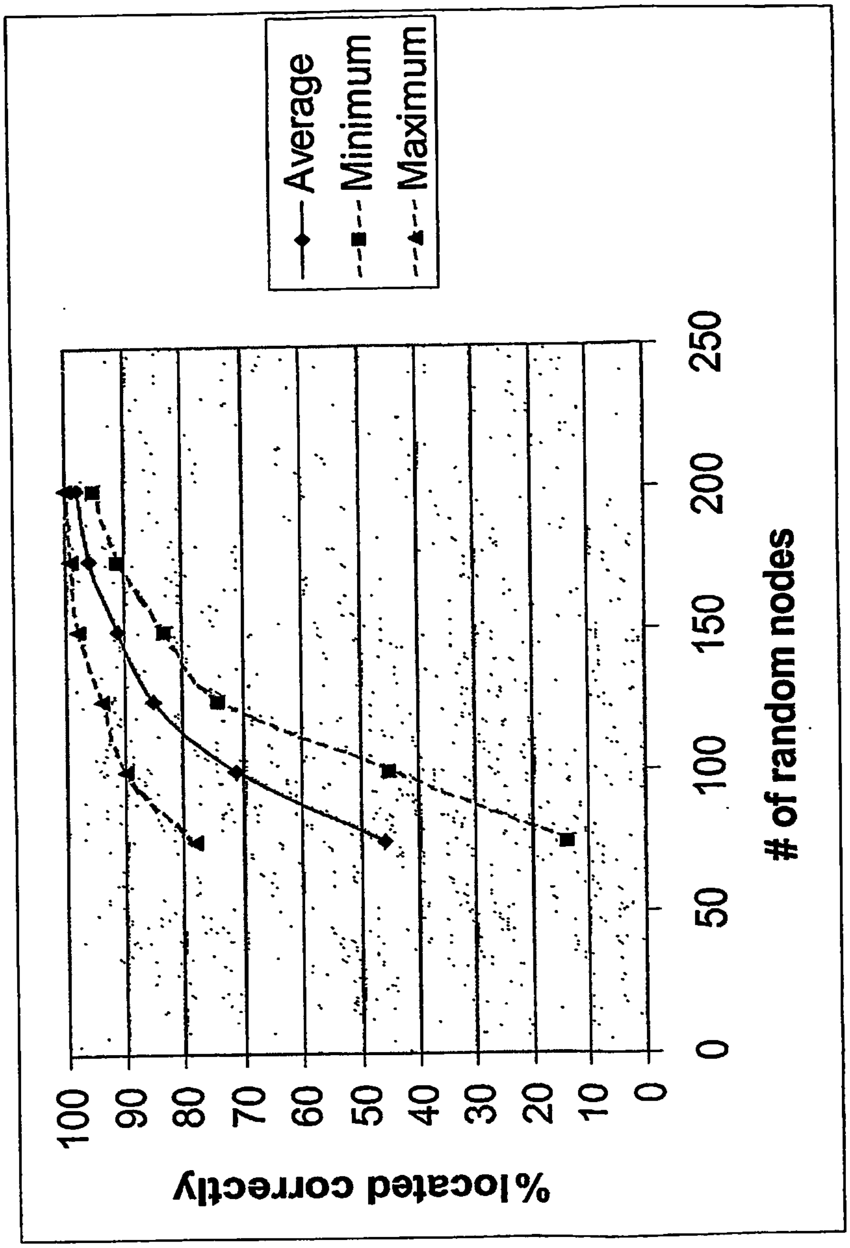
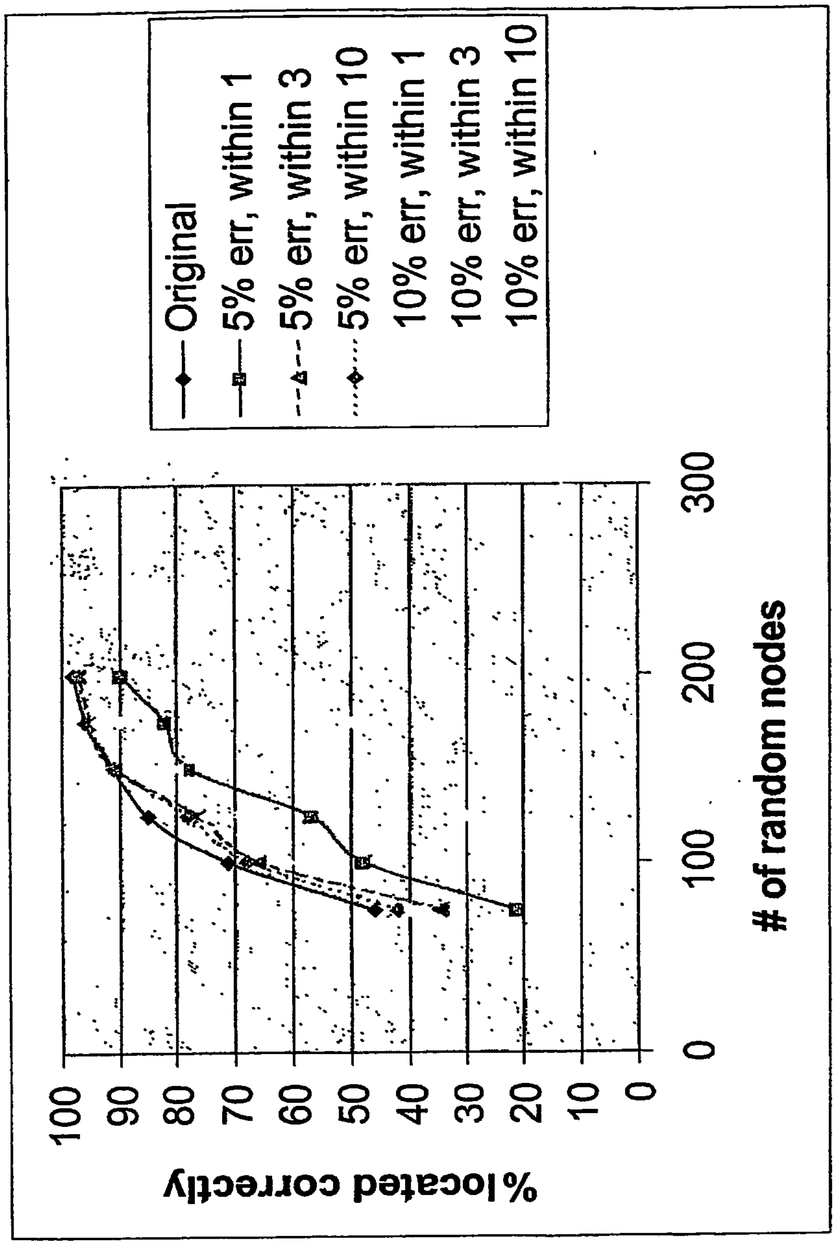


Figure 7



Average
Minimum
Maximum
% located correctly
100
90
80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
# of random nodes

Figure 8



% located correctly
100
90
80
70
60
50
40
30
20
10
0
0
100
200
300
# of random nodes
Original
5% err, within 1
5% err, within 3
5% err, within 10
10% err, within 1
10% err, within 3
10% err, within 10

Figure 9

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- [x] BLACK BORDERS
- [ ] IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- [x] FADED TEXT OR DRAWING
- [ ] BLURRED OR ILLEGIBLE TEXT OR DRAWING
- [x] SKEWED/SLANTED IMAGES
- [ ] COLOR OR BLACK AND WHITE PHOTOGRAPHS
- [ ] GRAY SCALE DOCUMENTS
- [x] LINES OR MARKS ON ORIGINAL DOCUMENT
- [ ] REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- [ ] OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.